

整理番号:

発送番号:630230 発送日:平成20年10月21日

1

引用非特許文献

特許出願の番号

特願2006-521892

作成日

平成20年10月10日

作成者

木村 雅也

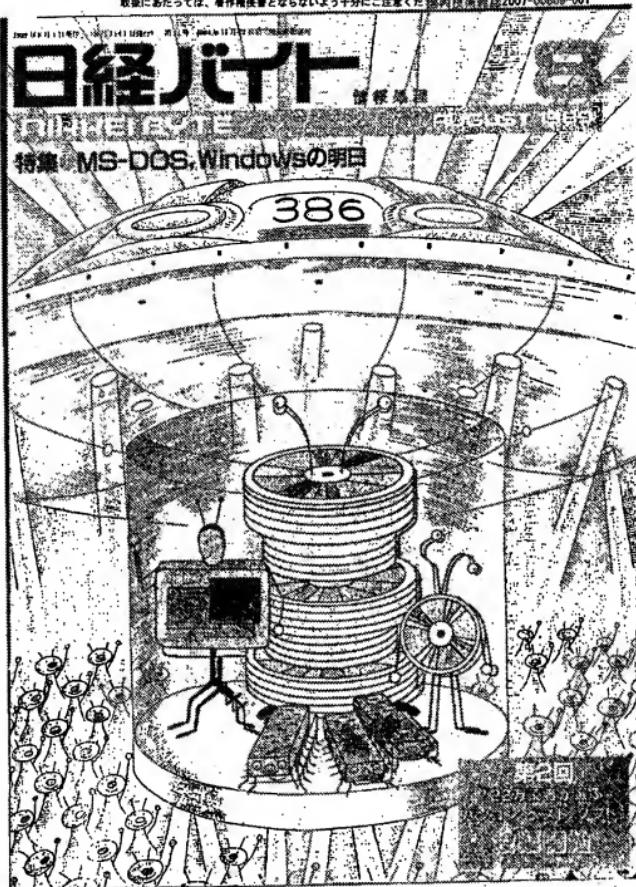
3980 5100

発明の名称

プランクを移送するためのシステム及び方法



本複製物は、特許庁が著作権法第42条第2項第1号の規定により複製したものとし
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。権利者:日経BP社 2007-000002-001



themes

第4部 高速ファイル

EMSメモリとキャッシングの活用で
ファイル性能を大幅に改善

ピエット・チュン

- EMSのPC-DOS 4.0は、ファイル・アクセス性能を大幅に向上させるために、四つの新機能を備えた。
- EMSメモリに最大4万個のシステム・ファイルを同時に読み込める「ラージ・バッファ」。
- シーケンシャル・アクセス性能向上のための「ルックヘッジ」・「バッファ」。
- ファイルを開く速度を高速化する「Fast Open」。
- ハードディスクにアクセスする速度を向上させる「Fast Seek」である。
- EMSの設計者自身が、これらの仕組みを解説するとともに、性能向上の実際データを示す。(本誌)

PC-DOSは、その発売からまだ7年しか経過していないが、最も普及しているOSである。IBM PCのハードウェアとアプリケーションの多様化に伴い、DOSも機能を向上した。最初のバージョン4.0では、ファイル性能を向上させるために四つの機能を実現した(バージョン3.0でも一部実現)。

第1の機能は、EMSメモリを使った「ラージ・バッファ」だ。大きなファイルのランダム・アクセス性能の向上が目的である。ユーザーは最大1万個のシステム・ファイルを操作できる。

第2の機能は、シーケンシャル・ア

クセスのための「ルックアヘッジ・バッファ」だ。この目的は、シーケンシャル・アクセスディスクの「アクセス回数を減らすこと」である。ランダム・アクセス性能を低下させることなしに、シーケンシャル・アクセス性能を向上させることである。ユーザーは最大1万個のルックアヘッジ・バッファを指定できる。

第3の機能は「Fast Open」だ。これはファイル・ディレクトリ・エントリのキャッシングである。通常にファイルを開く速度を向上させる場合に効果を

を発揮する。

第4の機能は「Fast Seek」だ。これはファイル・クラスター倍数のキャッシングである。単にディスク上のファイル・クラスター情報をキャッシュするだけでなく、クラスターの連続性など、より複雑の高い機能も含む。クラスターが組織したファイルのランダム・アクセスでは、大幅に性能が向上する。

Fast OpenとFast Seekは、ともにキャッシング内部をLRU方式¹で管理する。また、メモリをより効率的に使用するよう設計した。

ランダム・バッファ

本誌のフォーマット・チュン氏(Pete Che)は、米HDS社(ワシントン州リトルボローリーフィング)システム開発部、IBM PCの専門家である。IBM PCの開発と、PC DOS プログラムの開発に携わる。彼の著書は「IBM PC用C言語入門」である。

EMS 4.0は、Lotus/Intel/Microsoft社の発売する基板メモリ仕様の規格である。IBM PC用C言語入門

とで600 Kバイト以上のメモリにアクセスできる。EMSドライバの実装は、640 K-1 Mバイトの間の実装用アドレス空間に物理ページ(16 Kバイト単位)をいくつか割り当てる。これらをEMS

本資料は、特許庁が著作権法第42条第2項第1号の規定により複製したものです。
収録にあたっては、著作権侵害とならないよう十分にご注意ください。

themes

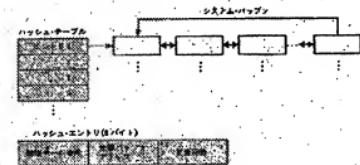


図1 ラージ・バッファの構造。ハッシュ・テーブルはコンピューションル・メモリ(既存の8088マイクロプロセッサー)に、バッファRAMメモリに格納されたハッシュ・テーブルの構造トリは、【物理ページ番号】、【物理・バッファ・アドレス】、【物理段番号】、【未ログイン・バッファ】で構成される。このうち、物理ページ番号は16ビットで、物理・バッファ・アドレスは、既存リスト構造のアドレス(16ビット)と、物理・バッファ・アドレス(16ビット)の計32ビットである。空翻番号は既存のハッシュ・テーブル

生3) just recently generated. サーバ上に書き込まれた場合、該当する物理ページはまだ読み込まれていない(バッファ読み込み未)。ただし、既存アドレス(16ビット)は、キャッシュに読み込まれる。

メモリ内の論理ページ(16Kバイト単位)にマップすること。

DOS 4.0では、ファイル・アクセスの効率化のために、一つの物理ページ(物理ページ番号256)を複数、システム・バッファをEMSメモリに確保できるようにした。これがラージ・バッファである。

以前のバージョンのDOSと同様に、config.sysファイル中のbuffersコマンドの最初のパラメータには、ファイル入出力のためのシステム・バッファ数を指定する。ここで新しい/オプションを併せて指定すると、最大1万個のバッファをEMSメモリに確保できる。1万個のバッファは、約5MバイトのEMSメモリに相当する。/オプションを有効化した場合は、バッファは最大32個に制限され、コンバージョンナル・メモリ(従来の540Kバイト以下の主記憶)中に確保される。

ハッシングとLRU方式で ランダム・アクセス性能を向上

ラージ・バッファは、二つの制限条件に縛られて設計された。

一つは、物理ページの大きさが16Kバイトであるため、一度にログセク用意なバッファが16Kバイトだけであること。

もう一つは、指定されたバッファ数が制限であっても、バッファを一定時間内に満足して供給できることだ。

この二つの制限条件を満たすために、図1に示すようなハッシュ・テーブルを用いた。

ハッシュ・テーブルの各エントリは、

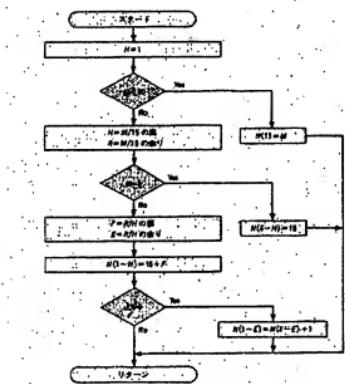


図2 ハッシュ・エントリに基づく既存のバッファを管理するアルゴリズム
H0: ハッシュ・リストヘッダー
H1: ハッシュ・リストヘッダー
H2: ハッシュ・リストヘッダー
H3: ハッシュ・リストヘッダー

本複製物は、特許庁が著作権法第42条第2項第1号の規定により複製したもので、取扱いにあたっては、著作権侵害とならないよう十分にご注意ください。

本複数物は、特許庁が著作権法第42条第2項第1号の規定により複製したもので、
取扱にあたっては、著作権侵害とならないよう十分にご注意ください。

図5 ランダム・アクセスでのルックアップ・バッファの検索。ハード・ディスク上の256ミバイト・ファイルに対してランダムなセクタを読み書きする1万回数回実行した。PS/2モデル80で実現した。

これは、ファイバのほとんどどのセクタがバッファ内に格納され、ディスク入出力なしでデータにアクセスできるからだ。

図6 ルックアップ・ヘッド・バッファ

ルックアップ・ヘッド・バッファの目的は、ランダム・アクセスの際に最終のセクタを読み出しておくことで、ディスク読み出しの回数を減らすことだ。ルックアップ・ヘッド・バッファのBMPは、config.sysファイルのbufferコマンドの2番目のパラメータで指定する。最大512のルックアップ・ヘッド・バッファを指定できる。

例えば、このパラメータを3とするとき、DOSはコンパインショナルメモリ中に3箇のルックアップ・ヘッド・バッファを確保する。この状態でディスクからセクタ10を読み出さ場合、DOSは次回にはセクタ100, 101, 102の3箇のセクタを、既て読み出しておく。こうしておけば、次にセクタ101の読み出しを要求されたとき、ディスク人出力なしでルックアップ・ヘッド・バッファからデータを取り出せる。

ランダム・アクセスへの影響はない。

DOSは、小容量ルックアップ・ヘッド・バッファによってランダム読み出しの性能が低下するのも予測する。この性能低下は大体のような状況なのだ。読み出し要求を受けたたびに、現在求められているセクタを当(Cドライブ)と、以前に要求されたセクタを毎(Pドライブ)とすると、そして、Cドライブからセクタを読み出し、そうでなければセクタCだけを読み出す。

図5はPS/2モデル80で実現した。

図5 ランダム・アクセスでのルックアップ・バッファの検索。ハード・ディスク上の256ミバイト・ファイルに対してランダムなセクタを読み書きする1万回数回実行した。PS/2モデル80で実現した。

これは、ファイバのほとんどどのセクタがバッファ内に格納され、ディスク入出力なしでデータにアクセスできるからだ。

図6 ルックアップ・ヘッド・バッファ

ルックアップ・ヘッド・バッファの目的は、ランダム・アクセスの際に最終のセクタを読み出しておくことで、ディスク読み出しの回数を減らすことだ。ルックアップ・ヘッド・バッファのBMPは、config.sysファイルのbufferコマンドの2番目のパラメータで指定する。最大512のルックアップ・ヘッド・バッファを指定できる。

例えば、このパラメータを3とするとき、DOSはコンパインショナルメモリ中に3箇のルックアップ・ヘッド・バッファを確保する。この状態でディスクからセクタ10を読み出さ場合、DOSは次回にはセクタ100, 101, 102の3箇のセクタを、既て読み出しておく。こうしておけば、次にセクタ101の読み

図5 ランダム・アクセスでのルックアップ・バッファの検索。ハード・ディスク上の256ミバイト・ファイルに対してランダムなセクタを読み書きする1万回数回実行した。PS/2モデル80で実現した。

これは、ファイバのほとんどどのセクタがバッファ内に格納され、ディスク入出力なしでデータにアクセスできるからだ。

図6 ルックアップ・ヘッド・バッファ

ルックアップ・ヘッド・バッファの目的は、ランダム・アクセスの際に最終のセクタを読み出しておくことで、ディスク読み出しの回数を減らすことだ。ルックアップ・ヘッド・バッファのBMPは、config.sysファイルのbufferコマンドの2番目のパラメータで指定する。最大512のルックアップ・ヘッド・バッファを指定できる。

例えば、このパラメータを3とするとき、DOSはコンパインショナルメモリ中に3箇のルックアップ・ヘッド・バッファを確保する。この状態でディスクからセクタ10を読み出さ場合、DOSは次回にはセクタ100, 101, 102の3箇のセクタを、既て読み出しておく。こうしておけば、次にセクタ101の読み

図5 ランダム・アクセスでのルックアップ・バッファの検索。ハード・ディスク上の256ミバイト・ファイルに対してランダムなセクタを読み書きする1万回数回実行した。PS/2モデル80で実現した。

これは、ファイバのほとんどどのセクタがバッファ内に格納され、ディスク入出力なしでデータにアクセスできるからだ。

図6 ルックアップ・ヘッド・バッファ

ルックアップ・ヘッド・バッファの目的は、ランダム・アクセスの際に最終のセクタを読み出しておくことで、ディスク読み出しの回数を減らすことだ。ルックアップ・ヘッド・バッファのBMPは、config.sysファイルのbufferコマンドの2番目のパラメータで指定する。最大512のルックアップ・ヘッド・バッファを指定できる。

例えば、このパラメータを3とするとき、DOSはコンパインショナルメモリ中に3箇のルックアップ・ヘッド・バッファを確保する。この状態でディスクからセクタ10を読み出さ場合、DOSは次回にはセクタ100, 101, 102の3箇のセクタを、既て読み出しておく。こうしておけば、次にセクタ101の読み

本機関は、特許庁が著作権法第42条第2項第1号の規定により複数したもので、
取扱にあたっては、著作権者とならないよう十分に注意ください。

CHAPTER 6

次に fastopen コマンドの形式は、以下のキャッシュ・エンタリ。ただし、
以下の通りである。
fastopen <file> <opt>
ここで、<file>はキャッシュを開放するファイル
<opt>は、<file>を Fast Open のオプションで
マップメントする。<opt>は Fast Open

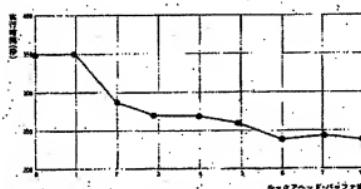


図6-1 シーケンシャル・アクセスでのルックアップ・バッファの経過。ハード・ディスク上の256ファイルに対して、シーケンシャルなセクタ読み書きを1万回実行した。Fast Open が用いられなかった。

作成コストの低減である。このコストでは、ハード・ディスク上の 256 K バイト・ファイルに対し、シーケンシャルなセクタ読み書きを 1 万回実行した。また、ランダム読み出し・検索までの下記の効率を評価するために、

前述のワージ・バッファの性能テストも実施した。その結果、ルックアップ・バッファはシーケンシャル・アクセス速度を大幅に向上させるが、ランダム・アクセス速度には影響しないことがわかった。

Fast Open

ディスク上のファイルは、それぞれ 256 バイトのディレクトリ・エントリで管理されている。ディレクトリ・エントリは、ファイルの属性や変更日付・時間・サイズ・ロケーションの組合せなどの情報を含む。ファイルのオープンを要求すると、DOS はそのディレクトリ・エントリをディスクからシステム・バッファに読み出し。その後の読み出しで情報を取得可能であるようにする。こうした方式を採っているため、毎ファイルに対してオープン・読み出し・書き込み・クローズを頻繁に行うアプリケーション・プログラム

では、アクセス回数が低下することがある。

第 1 に、32 バイトのディレクトリ・エントリを読み出するために、DOS はセクタ (通常 512 バイト) をまるごと読み出す必要がある。例えば、システムの書き込みバス名 (A:HWYCDなど) は、システム・バッファ中の 4K (32 × 1024) バイトを指す。その結果、システム・バッファの LRU 方式の結果が並び下る。

第 2 に、バッファ 数に制限があらため、ファイルをオープンした後で実行する読み出しと書き込みによってディ

レクトリ・エントリを含むバッファがブロックされる可能性がある。この場合、次の再オープンでは再度、ディスクを読み書きしなければならない。

DOS カーネルが利用する

主記憶常駐プログラム

こうした性能を認識するために、Fast Open を実装した。これは、主記憶常駐プログラムである。ディレクトリ・エントリを私有化してキャッシュすることで、DOS カーネルがディレクトリ・エントリに高速アクセスできるようになる。キャッシュ中のディレクトリ・エントリはブリード回避にしてあり、LRU 方式で管理する。確実なキャッシュ・エンタリを達成するため、fastopen コマンドの最初のパラメータで指定する。

DOS カーネルと Fast Open 用のインクルーズは、以下のように設計した (図 6)。

• Lookup: DOS カーネルは、ディスクからディレクトリ・エントリを読み出す前に、いつも Fast Open にルックアップ要求を発行する。この結果は、Found, Partially Found, Unfound の三つのいずれかになる。例えば A:HWYCD というバスを指定する。まずは Found だ。Fast Open は D のディレクトリ・エントリを返す。Partially Found なら、バス中のサブディレクトリ (例えば A, B, C, または D) のディレクトリ・エントリを返す。Unknown の場合は、DOS カーネルがディスクにアクセスして A, B, C, そして D のディレクトリ・エントリを検索し

本復元物は、特許庁が著作権法第42条第2項第1号の規定により複製したものです。
収後にあたっては、著作権侵害とならないよう十分にご注意ください。

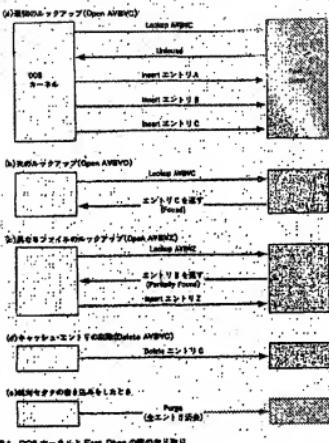


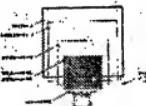
図1 DOSガードとFast Openの関係図

て次のInsertコマンドでそれらをキープする。キャッシュに入れておかねばならない。

- Insert: キャッシュにディレクトリエンタリを入れる。ファイルをマークエンタリを入れる。ファイルをマークエンタリと並べるなどを使う。
- Update: キャッシュからディレクトリエンタリを削除する。ファイルの削除やファイル名の変更の際に使う。
- Delete: キャッシュ内のディレクトリエンタリを削除する。ファイル名を削除する。

Fast Open: キャッシュ内のディレクトリエンタリを更新する。ファイルの属性の変更などの際に使う。

目を守る、
数々のシールド機能を
結集しました。



アプロテクトスーパーは、多機能の機能を
合めたアプロテクトのアドバンスドな上位
版のアプロテクト。アドバンスドな機能が
アプロテクトの機能のないアプロテクトより
多くの機能の変更を可能としたのがアプロ
テクトの大きな特徴である。アプロテクト
は、Nレベルの深层次のサブディレク
トリから、NのNレベルの階層のサ
ブディレクトリに、ファイルを保護する
ものを10階層に及ぶ。ここで、Nは

高性能電脳版シールドフィルター
ア・プロジェクト

スーパー

Q 協賛ガス化工連続式発送
マツダガス・東京ガス・関西電力

10月発売予定

本複製物は、特許庁が著作権法第4・2条第2項第1号の規定により複製したもので、取扱にあたっては、著作権侵害とならないよう十分にご注意ください。

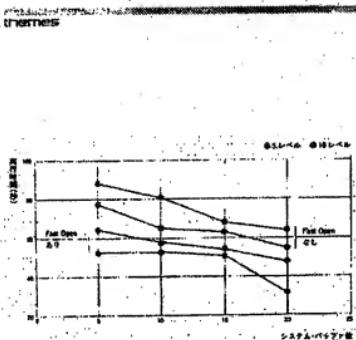


図7-28 エントリのFast Openキャッシュの結果。#レベルの直下のサブディレクトリから、次の#レベルの直下のサブディレクトリに、ファイルを保存するのを10回繰り返した。ここで、#はまとめて10である。各サブディレクトリは8GBバイト・ファイルを10個までいる。P4/2モデルで実験した

または10である。各サブディレクトリは22Kバイト・ファイルを5個含んでいる。この結果は、Fast Openのサッシュ・エントリを20個にすると、…貰った性能向上が得られることを示している。

卷之三

IOSのディスクマガジンショーンの基本操作はスクロールである。これは、左側のセレクターに的操作する（スクロールはセレクタ既定ショットマップに決まる）。ユーザの選択は、ファイルは既定であります。他のタグは既定で認識しているように思われる。すなはち、ファイルは既定タグアリから始めて複数タグアリ、並んでタグアリと認識されてしまう。しかし、実際には、ファイルは既定で決まっているタグで検出されているとは限らない。例えば、記憶タグアリが物理タグアリと並んで記憶されている場合、物理タグアリと、論理タグアリと認識される場合がある。これは、タグアリと記憶タグアリとが複数タグアリにマッチされるからである。

使っているか。米国版のクライアントはどれか、などの情報をファイル「アロケーション・テーブル」(PAT) に記述する。PAT はディスク上にあり、各 FAT エントリは各クライアントと 1:1 に対応している。ファイルの先頭クライアントは、デリケートリエントリに記述し、FAT エントリに記述するクライアントである。後続のクライアントはこの FAT エントリを次々にチェックすることで管理する。DOS は、ファイルの最初のクライアント、即ちクライアントをたどることで既存のファイル「デリケーション」を計算するわけだ。

この方法では、Fat Open の場合と場合に由来するに見えることがある。

第1に、FATセクタ(FATを含むセクタ)が複数のファイルのクラスタ - チェインを含むことがあります。またクラスタ - チェインが複数のFATセクタにまたがることがある。前者の場合、一つのファイルのクラスタ - チェインをまとめるために、DOSはいくつものFATセクタをシステム・バッファで読み出さなければならない。このためにバッファ・スペースが無駄になります。システム・バッファのLRU方式の効果が低下する。

第2に、もしも発現クラスターが連続しているなら、相鄰となるクラスター間でオフセットを加算するだけで目的のクラスター番号を簡単に計算できる。この場合でも、クラスター・オーバーラップを物理的にたどるのは非常に手堅い。

クラスタの属性値で

シク速度を向上

こうした操作を実現するにあたり、Fast Seekを実現した。クエリ・チーフの実装が通常方法で実行されるように、専用の操作でFast Seekを実現する。Fast Seekと通常点が多いので、Fast Seekはfastopenコマンドで一体化している。Fast Seekのためのキャッシュ、エンストリク等はfastopenコマンドF21の部分のパラメータで指定する¹⁴⁾。

Fast Seekのキャッシュ、エンストリク等は、ファイルの部分ごとに順序で実行しているが、という情報を含んでいない。この情報を順序実行機能リストの開頭にクエリリストと接続クエリリストに表示される。ファイルの順序実行が

本複製物は、特許庁が著作権法第42条第2項第1号の規定により仮託したもので、取扱にあたっては、著作権侵害とならないよう十分にご注意ください。

ATをAXに変える
AX-KIT。
プロサイトが
89年に贈る夢!!

開発価格と
実現技術の実現

主な機能
・8086/8087/8088/80386
・80486/80487/80488

■PMS86DX/80386-X(KIT版)

■PMS86DX-1(KIT)
■PMS86DX-1(KIT)
■PMS86DX-2(KIT)
■PMS86DX-3(KIT)
■PMS86DX-4(KIT)
■PMS86DX-5(KIT)
■PMS86DX-6(KIT)
■PMS86DX-7(KIT)

■PMS86DX/80386-X(AX版)

■PMS86DX-1(KIT)
■PMS86DX-1(KIT)
■PMS86DX-2(KIT)
■PMS86DX-3(KIT)
■PMS86DX-4(KIT)
■PMS86DX-5(KIT)

拡張機能付メモリ増設板(80386用)

■Fast Seek キャッシュ

■AX-KIT版
■Fast Seek キャッシュ

■AX-KIT版
■Fast Seek キャッシュ

■PMS86DX-1(KIT)
■PMS86DX-1(KIT)
■PMS86DX-2(KIT)
■PMS86DX-3(KIT)
■PMS86DX-4(KIT)
■PMS86DX-5(KIT)

■AT/AT-X版

■PMS86DX-1(KIT)
■PMS86DX-1(KIT)
■PMS86DX-2(KIT)
■PMS86DX-3(KIT)
■PMS86DX-4(KIT)
■PMS86DX-5(KIT)

■SOFT BOOK(AX-800)

■M1.0(AX-800)
■M1.0(AX-800)
■M1.0(AX-800)
■M1.0(AX-800)
■M1.0(AX-800)
■M1.0(AX-800)
■M1.0(AX-800)
■M1.0(AX-800)
■M1.0(AX-800)
■M1.0(AX-800)

■プロサイト版

■PMS86DX-1(KIT)
■PMS86DX-1(KIT)
■PMS86DX-2(KIT)
■PMS86DX-3(KIT)
■PMS86DX-4(KIT)
■PMS86DX-5(KIT)

■プロサイト版
■Fast Seek キャッシュ

■Pc プロサイト板式会社

■PC PHONE CORP

〒100 東京都千代田区神田駿河台1-1-10ルーム
03-5556-8181 FAX 03-5561-6134
〒780 三重県伊勢市伊勢町1-1-1
HONDA-133-3644 FAX 059-771-9611

themes

図1：Fast SeekのFast Seekキャッシュの効能。テスト内蔵は図と同じランダムアクセス

● Truncate：キャッシュからファイルのクエスト情報を削除する。ファイルを小さくした場合などに使う。

● Purge：キャッシュ内のすべてのクエスト情報を削除する。ディスクのフォーマットや、熱対セクタの書き込み直後 (Int 25h) がおった場合などに使用する。

図1はFast Seek性能のテスト結果である。テスト方法は、前述のデータ・ベース・プログラムの操作と同一である。この結果は、10秒のキャッシュ・エンタリを削除すると、一度約10秒程度ができる事を示している。システム・バッファが10回以上ある、Fast Seekを実行する回数の性質 (45回) を考慮するために、さらに60回ものバッファが必要だ。これは30KBのメモリに相当するが、Fast Seekを使わなければ5.4KBバイトのメモリで済む。

こうした結果は、すべてのファイル・クエスト情報をセーブするための十分なキャッシュ・エンタリをもつれば

一般に用意できる。結果の場合は、ファイルが大体の存続時間であり、ディスク上にまだらの分配されている場合だ。この場合、ランダム・アクセスを行なうと、LRU機制が効かず、Fast Seekは有効的な削除ができないくなる。こうした状況を解消する方法の一つは、キャッシュ・エンタリの利用回数を減らすまで、Ram Scanを一時的にシャットダウンすることである。◆◆◆◆◆

以上、DOSの両つの新しい機能を説明した。これらのおかげで、config.sysファイルのbuffersコマンドや、fatopenコマンドを削除することが可能となり、リカバリ用の領域を削除することができるようになった。同じコマンドで、リカバリ用のプログラムをコンピュータで実行した場合でも、ファイル・タスク処理が実行したことには気づくだらう。

(BYTE, © 1989 McGraw-Hill, Inc.)

お読みの方 209

10-006. Windows の実力：高速ファイル一覧 1000 BYTE/AUGUST 1989